logtrust

Logtrust Scalability Test

Logtrust ran a series of tests to show how a Logtrust environment sized for 5TB - 6TB of data ingestion a day would perform when subjected to data rates in the ranges of 20TB, 30TB, and 100TB.

This is an extreme stress test made to simulate the potential huge spikes of data a customer might see during new product introductions and other events. For the purposes of this test, the load was arbitrarily set at 5TB/day and 145K events per second (EPS).

The Test Setup

The goal was to evaluate both the collection and analysis performance of the Logtrust platform during three high load stress tests: Data Streams, Query Load, and Logtrust Setup.

Data Streams

Three different event streams were sent during this test:

- 500k EPS
- 1M EPS
- 3.5M EPS

Data arrival was modeled to simulate real-world load and variability. Event size averaged 340 bytes. Events generated were stored in the table **test.keep. free.**

QUERY LOAD

To analyze query performance, queries were run across the full data set from the collecting stress test.

- One query grouped events by message field every 10s and counted the number of events.
- A second query searched for events containing the word “sasquatch”.
- Additionally, through the API, sparse events were generated and added to the same tables. These sparse events were analyzed by the second query and displayed, also via the API.

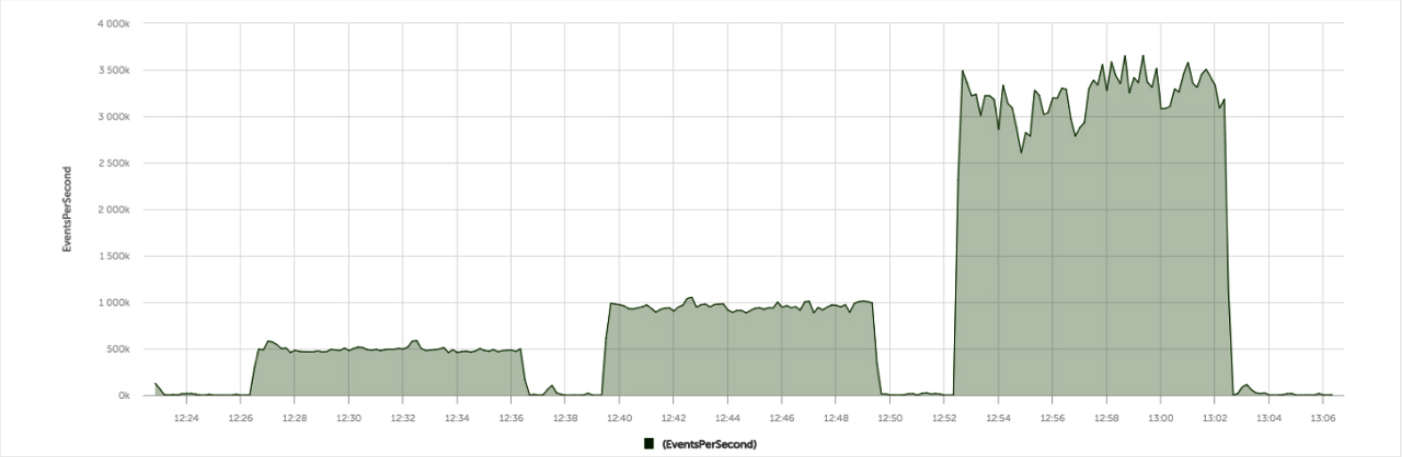

```
select period(eventdate,10*second()) as p, message, count() as Events from test.
keep.free where client = “sasquatchcis” and eventdate >= now() group by p, message

select * from test.keep.free where client = “sasquatchcis” and message ->
“sasquatch” and eventdate >= now()-30*minute()
```

LOGTRUST SETUP

For this test, Logtrust used six standard data nodes in our AWS-based cloud. All other services (UI, load balancing, etc...) used cloud infrastructure shared across our other customers. This is a typical sizing for a 6TB daily data rate.

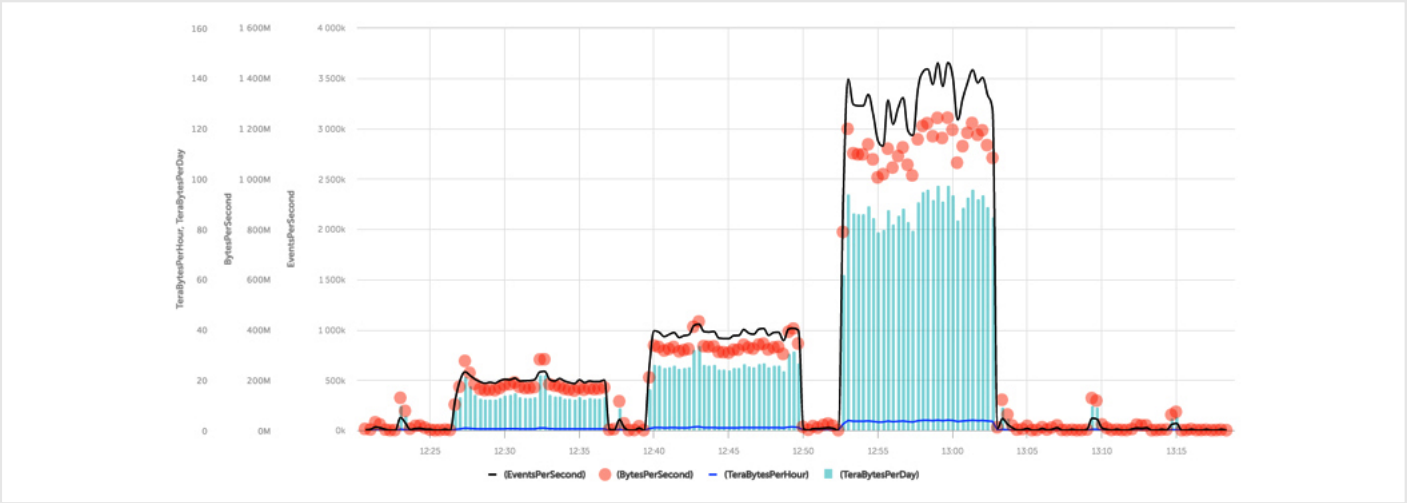
Data Collection

This graph shows the total number of events collected during the test scenarios. Even in these extreme test scenarios, not a single event was dropped by Logtrust. The variations seen in the event graphs show the variation in the data arrival from the simulation model used for these tests.



LogTrust Events Collected

The following table and graph show the resulting data rates per day and hour, based on the event streams used during the stress tests.

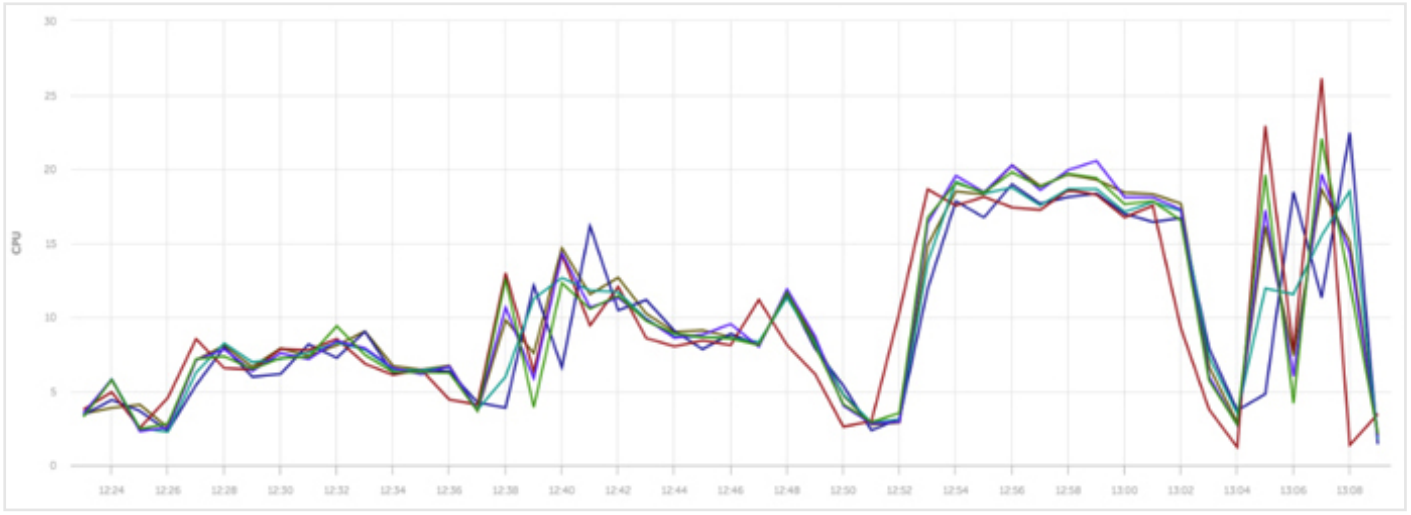


Event and Data Ingestion Rates

Events Per Second	Total Volume Per Hour (TB/hour)	Total Volume Per Day (TB/day)
500,000 EPS	0.55 TB - 0.93 TB	12.55 - 22.15 TB
1,000,000 EPS	1.02 TB - 1.42 TB	24.43 - 34.01 TB
3,500,000 EPS	3.32 TB - 4.07 TB	79.64 - 97.67 TB

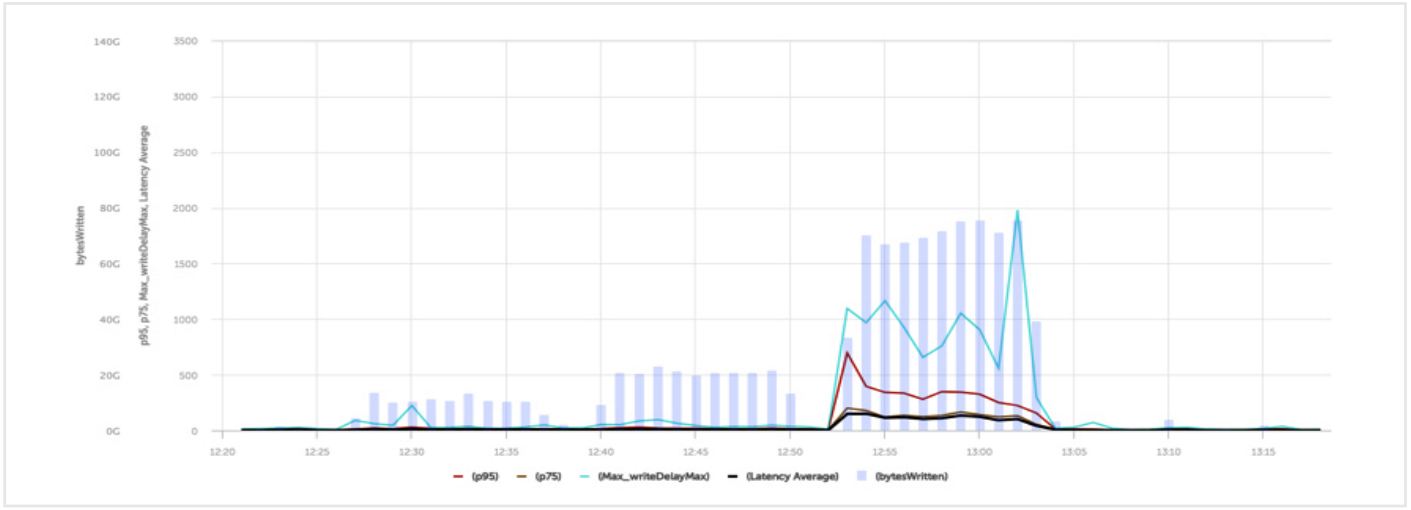
Once again, the variation in data rates is due to the variation in the data arrival from the simulation model used for these tests.

At the same time events were collected, CPU and write latency were monitored.



CPU Usage Across All 6 data nodes

At maximum workload (12:58), the average CPU load of the data nodes was 20%. Under this load, data nodes not only ingested all 3.8M events, but also performed post-ingestion tokenization on data already written to disk.



Write Latency

In Logtrust, data is available for query as soon as it is written to disk. Write latency is therefore a critical metric, as it indicates the maximum time it will take for data to be available for analysis.

At maximum collection, maximum write latency was two seconds with an average time of 133.66ms. Performance at the 95th percentile was 343.65ms and 165.25ms at the 75th percentile, writing at 8.89 Gbits/s.

CONCLUSION

This test shows that a Logtrust system sized for 5TB/day is able to handle a load test of a 97TB/day data stream (19x) without dropping a single event. Further, the load and write latency behaviors of the system, even at these high stress levels, provide sub-second access to data as it is streaming into the system, and the full capability to tokenize the data after it has been written to disk.

Query Performance

Query tests were performed across the full data set collected during collection testing. Both full scan queries and tokenized queries were performed to test a variety of query types within Logtrust.

NON-TOKENIZED QUERIES

The first test leveraged the following query to calculate the total volume of events generated during the data collection test.

```
select count() from test.keep.free where client = “sasquatchcis” and today() <=
eventdate <= now()
```

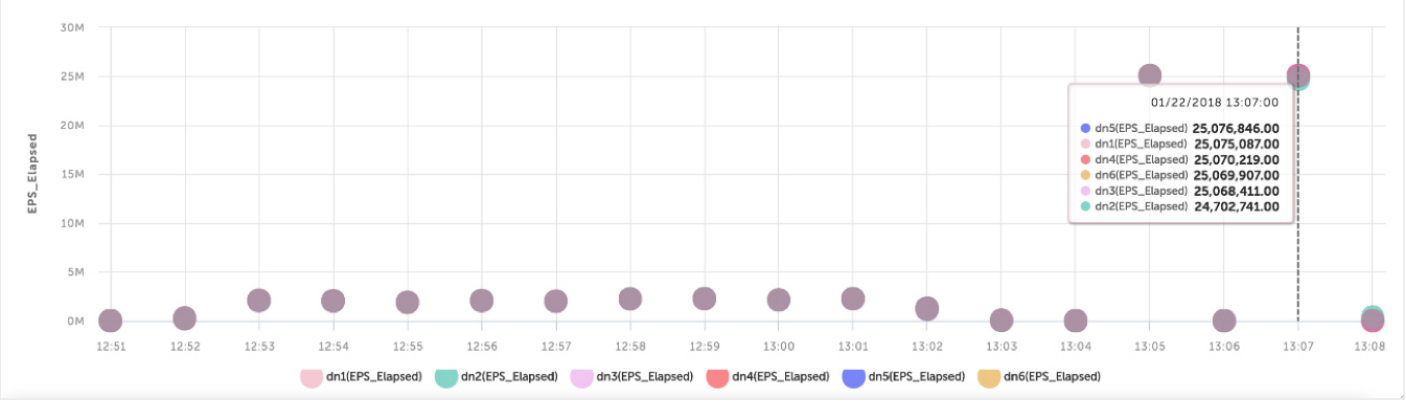
Query volume comprised a total of 6018451483 (> 6 billion events). The query was executed by making a full scan on the data set, which took **42.68s.** This was achieved using the six data nodes with a total of 16 query engines per machine running the query in parallel. Total load was **23,502,231 EPS per data node** (> 23M EPS per data node), a total of **1,468,889 events per query engine.**

A second query that searched for a token was also performed via a full scan on the data set. This query was performed without using tokenized data.

```
select * from test.keep.free where client = “sasquatchcis” and message ->
“sasquatch” and today() <= eventdate <= now() pragma index.cache.enabled:false’
```

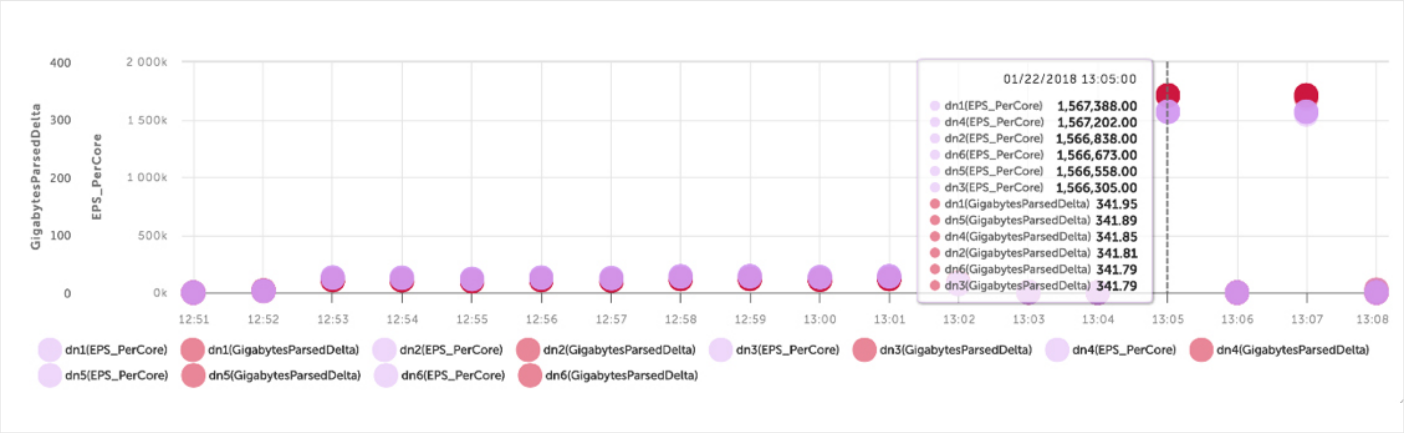
This query analyzed the same volume of events as the first query (> 6 Billion) and returned in **43.9s.**

In the following two graphs, the performance of each of the two previous queries is measured. Performance per data node shows the total information processed. Each data node analyzed 341 Gbytes of data at a rate of 1.5M EPS.



Write Latency

A total of > 25M EPS were queried per data node.



Tokenized Queries

A third query was performed that leverages tokenized data within Logtrust. It is identical to the second query across the same data set.

```
select * from test.keep.free where client = “sasquatchcis” and
toktains(message,”sasquatch”) and today() <=  eventdate <= now()
```

Total execution time was 0.466s, displaying the same results as the previous query but with a 95x improvement in speed to analyze > 6 Billion events. In this query, the CPU of the data nodes did not rise noticeably, due to the short time the query needed to run.

CONCLUSION

This test demonstrates Logtrust’s ability to do rapid full table scans when data is not tokenized, and shows that tokenized queries can perform even faster - up to 95x faster in our tests.

Summary

This test shows clearly that a Logtrust system sized for a moderate data load of 5TB can handle massive spikes and data surges at up to 100TB a day, without losing any data, while providing sub-second access to streaming data. Test results show the system provides predictable sub-second query latency, as well as the ability to analyze both real-time and historical data sets with a minimal hardware footprint and resource usage.